

"Express Mail" mailing label number:

EL 375 516 207 US

NETWORK PROCESSOR SYSTEM

Brian A. Petersen
Mark A. Ross

5

BACKGROUND OF THE INVENTION

Field of the Invention

This invention relates to the field of network processors, specifically network processors adapted to perform packet processing.

10 Description of the Related Art

In the data networking field there exists a long felt need to provide faster packet processing using fewer system resources and more efficient hardware. Those of ordinary skill in the art have long realized that a programmable processing system can be readily adapted to provide packet processing. However, such systems are typically
15 implemented in custom or semi-custom application specific integrated circuits (ASICs) which are difficult and costly to develop and produce. Furthermore, such ASICs are not readily changeable in the event that packet configurations, processing requirements, or standards change over time.

What is needed is a rapidly adaptable packet processing system able to be
20 easily configured to perform a wide range of packet processing tasks without redesign or reconstruction of the processor system hardware itself.

SUMMARY

Presently disclosed is a general purpose, software-controlled central processor augmented by a set of task specific, specialized peripheral processors (simply referred
25 to as "peripherals"). The central processor accomplishes its software-determined

functions with the support of the peripheral processors. Peripheral processors may include but are not limited to a packet parser, which provides the central processor with a numerical summary of the packet format; a packet deconstructor, which extracts designated fields from the packet, the positions of which are determined by the central processor according to the packet format; a search engine, which is supplied a lookup index by and returns its results to the central processor; and a packet editor which modifies the packet as determined by the central processor using the previously-identified information from other peripherals.

At each step in the use of this network processor system, the central processor has an opportunity to intervene and modify the handling of the packet based on its current interpretation of peripheral processor results. The programmable nature of the central processor and the peripheral processors provides the system with flexibility and adaptability. Rather than having to modify a circuit or system design in an ASIC or other complex hardware device, new packet processing applications may be accommodated through the development of new software and its deployment in the central and/or peripheral processors.

BRIEF DESCRIPTION OF THE DRAWINGS

The present disclosure may be better understood and its numerous features and advantages made apparent to those skilled in the art by referencing the accompanying drawings.

Figure 1 is a high-level block diagram of the central processor/peripheral processor architecture according to one embodiment of the present invention.

Figure 2 is a flowchart of the sequence of events by which a packet is processed according to one embodiment of the present invention.

The use of the same reference symbols in different drawings indicates similar or identical items.

DETAILED DESCRIPTION

Architecture

The network packet processor system of one embodiment of the present invention comprises a central processor (CP) and a set of peripheral processors (PP).

5 In some embodiments of this architecture, the peripheral processors each communicate only with the central processor; they do not communicate with each other. In other embodiments, the PPs can share information either passed from the CP or derived within one or more PPs with other PPs. The CP acts as the coordinating and controlling processor while each peripheral processor performs specialized tasks
10 with high efficiency. The advantage of this architecture is that the individual processor (CP and PP) workflows, tasks, and functions are completely modularized and configurable by appropriate processor programming.

Figure 1 shows a high-level block diagram of the present central/peripheral processor system architecture 100 for packet processing. Central processor 110 receives packets through any of a number of means well-known in the art. central processor 110 performs, in some embodiments, preliminary format checking, e.g., checksum validation, and passes the packet or parts of the packet to one or more peripheral processors for additional work. central processor 110 may pass data to one or more peripheral processors 120, 130, 140, and 150 in sequence, in parallel, or in a
20 pipelined fashion.

Central processor 110 is a general purpose programmable processor, such as (but not limited to) an embedded processor core available from Tensilica, Inc. of Santa Clara, California or Advanced RISC Machines (ARM) Ltd. of Cambridge, England. In some embodiments of the present invention, the embedded core forming
25 central processor 110 is part of an application specific integrated circuit (ASIC).

In one embodiment of the present invention, shown in Fig. 1, four peripheral processors 120, 130, 140, and 150 are employed. One of ordinary skill in the art will readily see that fewer or more PPs may be employed without deviating from the spirit

Sub B1 15

of the present invention. Accordingly, the present architecture is not limited to a certain number of peripheral processors.

Peripheral processors 120, 130, 140, and 150 may each be implemented independently in any form of processing module or ASIC known in the electronic arts.

5 For instance, any PP may be a collection of discrete, fixed (hard-wired) logic, a programmable or fixed state machine, a microsequencer or microprocessor, a stored program-controlled processor using either ROM or RAM storage or a combination thereof, or a general-purpose, fully programmable computer. Any implementation
10 packet processor system 100 overall. Accordingly, the present invention is not limited in the physical implementation of any PP.

In some embodiments of the present invention, central processor 110 and one or more PPs are contained in the same ASIC.

Sequence Of Events

15 In the embodiment of Fig. 1, the four PPs are packet parser 120, packet deconstructor 130, search engine 140, and packet editor 150. Each performs specific functions at the request of central processor 110 and returns results to central processor 110.

20 Packets are received and delivered simultaneously to packet parser 120. A buffer (not shown) may also be employed to provide latency compensation, as is well-known in the art. Packet error detection code(s), such as the well-known cyclic redundancy check (CRC) field, are verified if present. Reception errors are flagged and included as part of a status word that is associated with the packet by packet parser 120.

25 The packet is deposited into a latency buffer primarily to allow a minimum amount of data to accumulate for address lookup purposes. The latency buffer makes the receive packet data available to packet deconstructor 130 and central processor 110 prior to the packet being stored in a central packet data buffer (not shown).

Packet parser 120 takes a quick look at the received packet and assigns a “vector” to the packet that indicates to central processor 110 in which of several categories (based on, e.g., packet formats) the packet belongs. A vector, as used here, is an identifying number or data field, such as simple byte code “0xF8” (F8 in hexadecimal notation). The vector can be one or more bits, bytes, or words. This provides central processor 110 a head start in the processing of the receive packet. Knowing the packet vector, central processor 110 knows where in the packet the fields of interest are located without having to examine the packet itself. This knowledge is stored in central processor 110, in one embodiment, using templates that indicate the desired fields for each vector, i.e., for each type of packet. Operationally, if the packet conforms to one of several expected formats as indicated by the vector, the appropriate processing template held within packet deconstructor 130 is selected by central processor 110. Packet deconstructor 130 executes the selected template by reading the required data directly from the latency buffer using pointers maintained by the latency buffer.

Packet deconstructor PP 130 delivers one set of selected fields to central processor 110 and accumulates a (possibly different) set of fields into a search argument that it delivers (in some embodiments) directly to search engine PP 140. In other embodiments, the accumulated search argument is delivered to search engine 140 via central processor 110.

In either event, the search argument is used to extract routing information, such as the destination port, MAC address, or IP address (as appropriate to the routing level of interest) from the routing data structures, which in some embodiments consist of tables. Various types of packet routing lookups can be performed by search engine 140, such as the well-known OSI Layer 2, Layer 3, and/or Layer 4 lookups. The search yields search results that are returned to central processor 110. Typically, only one of the lookups results in a destination determination; the layer 2 destination address lookup, in particular, determines which lookup identifies the packet’s next destination. Central processor 110 has the option of examining the search results and

modifying the destination returned by the lookups as necessary, in case of error or exception.

Editor PP 150 uses the information derived from parser 120, packet deconstructor 130, search engine 140, and central processor 110 to modify the packet (especially its header) in order to guide the packet to its next destination. This is the last step of the well-known routing/switching function performed by most packet processing systems.

Before the packet is forwarded by the switch/router, it is stored (buffered) in a packet data buffer (not shown). Such storage, including all necessary queuing, flow management, buffer management, retrieval and outbound (egress) forwarding and the like, may be accomplished by any of a number of means well-known in the packet processing and networking arts. Accordingly, packet storage (and subsequent retrieval) will not be further discussed herein.

Figure 2 is a flowchart of the sequence of events discussed above. Packet processing 200 begins with packet reception 210 and buffering 220 to accommodate latency. Packet parsing 230 is next accomplished to determine a packet vector by which the packet is internally identified.

Processing coordination and control 240 evaluates the packet vector and passes the packet (either directly or by reference) to packet deconstructing step 250. Packet deconstructing 250 deconstructs the packet into its constituent parts, e.g., header fields, quality of service (QoS) bits, packet data payload, etc. The results of deconstructing 250 are passed back to processing step 240 and, in some embodiments, directly to searching (lookup) step 260.

Lookup results from search step 260 are returned to processing step 240 where they are used to control packet editing step 270. The revised packet is then sent for storage and forwarding 299 by means well-known in the art.

At any time in process 200, processing step 240 may, upon evaluation of the results of any PP step 230, 250, 260, or 270, redirect or alter the processing scheme

according to its own (i.e., the central processor's) programming. Such redirection may occur, for instance, on an error or exception condition, such as the failure of a packet to pass a CRC check or the receipt of an illegal format.

Alternate Embodiments

5 While central processor 110 is described as a single, monolithic processor, nothing in the architecture of the present invention so limits its implementation. In particular, central processor 110 can be formed as an interconnected network or mesh of two or more processors acting in concert. These processors forming central processor 110 may be implemented in the same ASIC or other integrated circuit
10 device or on multiple ASICs or other integrated circuit devices. Such multi-processor implementations of a single processing function (such as that of central processor 110) are well-known to those of ordinary skill in the art.

Furthermore, while central processor 110 may be implemented as one or more interconnected processors, the peripheral processors, as a group, may also be
15 implemented in one or more "sets" of PPs in order to pipeline or parallel packet processing across multiple peripheral sets under the control of a single central processor entity. As with central processor 110, the above-described PPs may be implemented on one or more ASICs or other integrated circuit devices.

In a further alternative embodiment, central processor 110 and the peripheral
20 processors (in one embodiment, PPs 120, 130, 140, and 150) share a common set of registers in order to speed up data transfer between them and calculations using the same data. Some or all of the registers used by central processor 110 and all or some of the peripheral processors may be logically mapped to the same memory locations or otherwise shared by means long known in the computer and
25 microcomputer/microprocessor arts.

The order in which the processing steps of any embodiment of the present invention are performed is purely illustrative in nature. In fact, these steps can be performed in any order or in parallel, unless otherwise indicated by the present disclosure.

The method of the present invention may be performed in either hardware, software, or any combination thereof, as those terms are currently known in the art. In particular, the present method may be carried out by software, firmware, or microcode operating on a computer or computers of any type. Additionally, software embodying the present invention may be in the form of computer instructions in any form (e.g., source code, object code, interpreted code, etc.) stored in any computer-readable medium (e.g., ROM, RAM, magnetic media, punched tape or card, compact disc (CD) in any form, DVD, etc.). Furthermore, such software may also be in the form of a computer data signal embodied in a carrier wave, such as that found within the well-known Web pages transferred among computers connected to the Internet. Accordingly, the present invention is not limited to any particular platform, unless specifically stated otherwise in the present disclosure.

While particular embodiments of the present invention have been shown and described, it will be apparent to those skilled in the art that changes and modifications may be made without departing from this invention in its broader aspect and, therefore, the appended claims are to encompass within their scope all such changes and modifications as fall within the true spirit of this invention.